ACEEE Proc. of Int. Conf. on Advances in Power Electronics and Instrumentation Engineering, PEIE

# Delta Domain Modeling and Identification Using Neural Network

Suman Kumar Ghosh and Prasanta Sarkar
[1]UIT, Burdwan University, India.
Email: s.k.ghosh@hotmail.com
[2] NITTTR Kolkata

***Abstract*— This paper proposes modeling and identification of dynamical systems in delta domain using neural network. The properties of delta operator are used such as greater numerical robustness in computation and superior coefficients representation in finite word length in implementation and well ensured numerical conditioning at high sampling frequency. To formulate the identification scheme delta operator model is recasted into a realizable neural network structure using the properties of inverse delta operator.**

***Index Terms*— Delta Operator, Neural Network, System Modeling, System Identification**

## I. INTRODUCTION

In recent years, the delta operator *[1]*, *[2]* has been widely used to many areas in systems and control. The delta operator establishes a special rapprochement between analog and discrete dynamic models and allows for investigating the asymptotic behavior of discrete-time models as the sampling period converges to zero. Moreover, it has certain numerical advantages compared to the shift operator parameterization. It therefore, provides a unified framework for system studies, where continuous-time results can be achieved from the discrete-time description of the system. Since 1990 there has been an explosive growth in pure and applied research related to neural network. During this period multilayer feed forward network was introduced and found applications primarily in static systems. K. S. Narendra and K. Parthasarathy *[8]* have demonstrated that neural network (NN) could be used efficiently for the identification and control of nonlinear discrete-time systems modeled with shift operator. Numerous models for practical identification of nonlinear dynamical systems were proposed by them. Following the publication of this paper several researchers in the last two and half decades have further reinforced the research activities with wider applications and greater practical importance in the area of system modeling and. In this area, the research initiatives have been more concentrated on the application of NN for system identification mostly modeled with shift operator *[7]*, *[9] to [14]*.

This paper proposes NN modeling and identification of linear dynamical systems modeled with delta operator to capitalize the advantages of greater numerical robustness in computation and superior coefficients representation in finite word length in implementation and well ensured numerical conditioning at high sampling frequency. The paper is organized as follows. Section *II* provides a brief introduction to delta operator. In section *III*, modeling and identification procedure in delta domain is described. A brief introduction of the concept of multilayer neural network is illustrated in section *IV*. In section *V,* the computational framework of NN model for delta operator system is presented for identification. Different

*DOI: 02.PEIE.2014.5.505*

examples have been discussed with simulation results. Finally conclusion is drawn in section *VI.*

## II. The Delta Operator

The delta operator is defined in the time-domain as

$$\delta = \frac{q-1}{\Delta} \qquad \text{.....(1)}$$

where, $\Delta$ is the sampling period and $q$ is the forward shift operator. Operating $\delta$ on a differential signal $x(t)$ gives

$$\delta x(t) = \frac{x(t+\Delta) - x(t)}{\Delta} \qquad \text{.....(2)}$$

Similar relation exists in the complex domain as well. The transform operator $\gamma$ is related to $z$-transform by the following linear transform:

$$\gamma = \frac{z-1}{\Delta} \qquad \text{.....(3)}$$

where $z = e^{s\Delta}$ is the complex domain $z$-transform operator while $s$ is the Laplace transform operator.

In discrete time using delta operator, an n[th] order single input single output linear time invariant system *[also referred to as plant in the following sections]* is represented by *[3]*:

$\delta\, \underline{x}\,(t) = A_\delta .\, \underline{x}\,(t) + B_\delta .u(t)$ ,

$y(t) = C_\delta \,.\, \underline{x}\,(t)$ .....*(4)*

where $A_\delta$, $B_\delta$ and $C_\delta$ are matrices in delta domain state-space representation.

The independent variable $t = \begin{cases} k\Delta; & \Delta \neq 0 \\ t; & \Delta = 0 \end{cases}$,

$k$ is a positive integer and $\underline{x}\,(t) \equiv [x_1(t), x_2(t), \ldots\ldots\ldots\ldots, x_n(t)]^T$ is the state vector with $x_i(t)$s $[i=1,2,\ldots\ldots n]$, are state variables and $u(t)$ & $y(t)$ represent the system input and output of the system respectively. The vector $\underline{x}\,(t)$ denotes the state of the system at time $t$ and is determined by the state at time $t_0 < t$ and the input $u$ is defined over the interval $[t_0\ t]$. The output $y(t)$ is determined completely by the state of the system at time $t$.

## III. Modeling And Identification

Consider a second order single input single output linear time invariant plant whose delta transfer function is given by

$$M_\delta(\gamma) = \frac{A\gamma + B}{\gamma^2 + C\gamma + D} \qquad \text{...(5)}$$

*Equ. (5)* can be written as

$$\frac{y(t)}{u(t)} = \frac{A\gamma + B}{\gamma^2 + C\gamma + D},$$

*or.* $y(t) = - C\gamma^{-1} y(t) - D\gamma^{-2} y(t) + A\gamma^{-1} u(t) + B\gamma^{-2} u(t),$

*or.* $y(t) = \alpha_1 \gamma^{-1} y(t) + \alpha_2 \gamma^{-2} y(t) + \beta_1 \gamma^{-1} u(t) + \beta_2 \gamma^{-2} u(t).$

*where* $\alpha_1 = -C$, $\alpha_2 = -D$, $\beta_1 = A$ *and* $\beta_2 = B$.

Thus the output $y(t)$ can be expressed as sum of gama-inverse values of input u(t) and output y(t). Again consider a $n^{th}$ order single input single output linear time invariant plant, whose delta operator representation is given by *equ.(4)*. If the plant is controllable and observable then the state as well as output at any instant can be determined from linear combination of $n$ gamma-inverse values, $\gamma^{-1} y_p(t), \gamma^{-2} y_p(t), \ldots, \gamma^{-n} y_p(t)$ of $y_p(t)$ and $n$ gamma-
inverse values, $\gamma^{-1} u(t), \gamma^{-2} u(t), \ldots, \gamma^{-n} u(t)$ of $u(t)$ where $n$ is positive integer. Equation *(4)* can then be written as
$y_p(t) =$

$$\sum\nolimits_{i=1}^{n} [\alpha_i(\Delta).\ \gamma^{-i} y_p(t)\ ] + \sum\nolimits_{j=1}^{n} [\beta_j(\Delta).\ \gamma^{-j} u(t)\ ]$$

where $\alpha_i^T(\Delta)$ and $\beta_j^T(\Delta)$ are parameter vectors dependent on sampling period Δ. A neural network identification model for estimation of $\alpha_i(\Delta)$ and $\beta_j(\Delta)$ of the linear plant is shown in *Fig.1* .

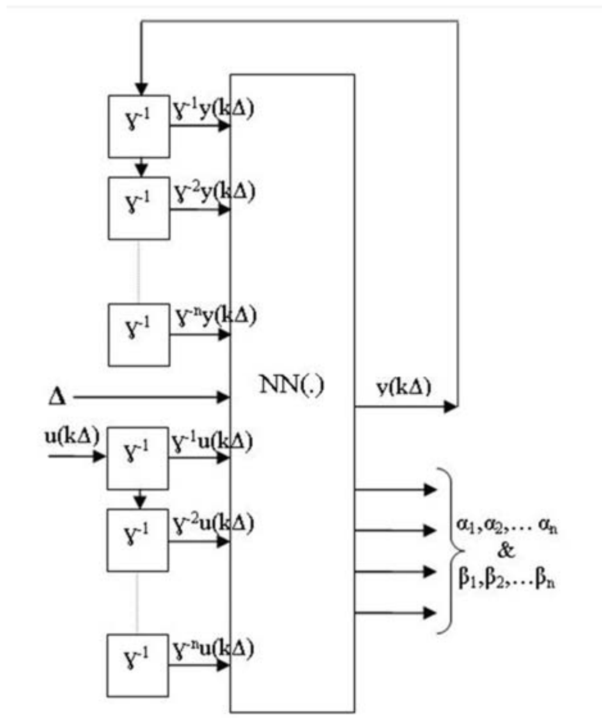


Fig.1: Neural Network Identification Model for Linear Plant

The gamma-inverse block used in the model structure can further be realized from *equ.(3)* into the structure shown in *fig.2*.

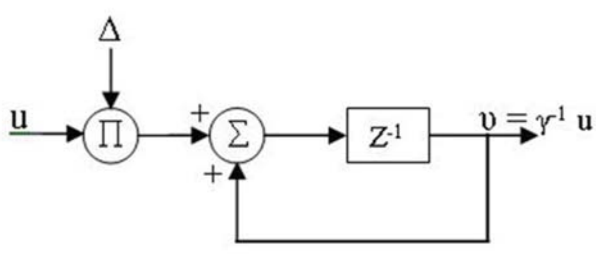


Fig2: Block Diagram Representation of Gama-inverse Operator

In the model plant output at the time instant $t$ depends both on gama-inverse values, $\gamma^{-i} y_p(t)$ $[i=1,\ldots,n]$ of $y_p(t)$ and gama-inverse values, $\gamma^{-j} u(t)$ $[j=1,\ldots,n]$ of $u(t)$.

IV. MULTILAYER NEURAL NETWORK

Multilayer neural network has apparent features like learning capabilities, inherent approximation capabilities, massive parallelism and very fast adaptability which enable them to be used as a powerful tool for modeling linear as well as nonlinear functions. One advantage of multilayer neural network is that the complexity of unknown function can be handled by increasing the number of neurons in each layer without much increase in computational time of the network due to parallel processing capability of neurons. The structure of a multilayer feed forward neural network containing two hidden layers is shown in *fig.3*

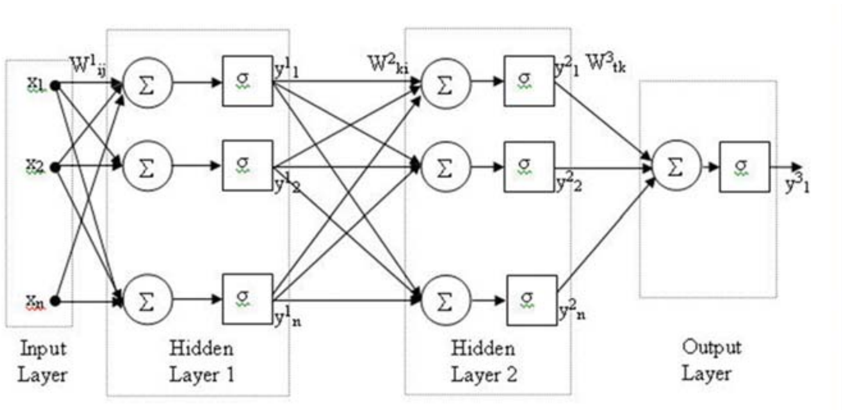


Fig3: A Multilayer Feedforward Neural Network with Two Hidden Layers

The output of the $i^{th}$ element (neuron) in layer $l$ is given by

$$y_i^l = \sigma\left( \sum_{j=0}^{n} w_{ij}^l y_j^{l-1} + w_{i,0}^l \right),$$

where $[w_{ij}^l]$ are the weights associated with the $i^{th}$ neuron at the $l^{th}$ layer and the function $\sigma(.): R \rightarrow (-1,1)$ is a monotone continuous function. Hyperbolic tangent function; *tanh(.)* is an example of such function.

Throughout this paper a multilayer network having $n_0$ number of neurons in input layer, $n_1, n_2, \ldots\ldots n_{l-1}$ no. of neurons in $1^{st}$, $2^{nd}$, …*and* $(l-1)^{th}$ hidden layers and $n_l$ no. of neurons in output layer is represented by $NN_{n1,n2,\ldots nl-1}$

A neural network, as defined above, represents a specific family of parameterized maps. If there are $n_0$ input elements and $n_l$ output elements, the network defines a continuous mapping $NN(.)$: $R^{n0} \rightarrow R^{nl}$.

The back propagation algorithm which performs gradient descent, provides an efficient method to train a multilayer feedforward neural network *[4]* to approximate a given continuous function over a compact domain. In this paper multilayer backpropagation neural network is used for modeling linear time invariant dynamical plant capitalizing its approximating property *[5]* of any continuous function on a compact set in a very precise and satisfactory sense.

V. SIMULATION RESULTS

For simulation purpose we have considered a second order linear plant having following model structure:

$y_p(t) = f_l[\gamma^{-1} y_p(t), \gamma^{-2} y_p(t), \gamma^{-1} u(t), \gamma^{-2} u(t)]$, $t = k\Delta$ for $\Delta \neq 0$; *k* is a positive integer.

Where the function $f_l(.)$ is a linear function of gamainverse values of $y_p(t)$ and $u(t)$.

For this example the unknown function is assumed to be of the form

$$f_l(.) = a\,\gamma^{-1} u(t) + b\,\gamma^{-2} u(t) - c\,\gamma^{-1} y_p(t) - d\,\gamma^{-2} y_p(t)$$

where a, b, c and d values are calculated from the continuous time s-domain transfer function

$y(t)/u(t) = [1.8868s + 0.5314] / [s^2 + 1.2217s + 0.5314]$

and their values are given in *table-I* for different values of $\Delta$.

The function $f_l(.)$ is estimated by a multilayer neural network $NN_{12,14}$ having two hidden layers of twelve and fourteen neurons respectively. The number of neurons in the hidden layer is chosen keeping in mind that the

TABLE-I: DELTA TRANSFER FUNCTION COEFFICIENT VALUES DERIVED FROM A CONTINUOUS-TIME MODEL FOR DIFFERENT Δ.

| Δ | a | b | c | d |
|---|---|---|---|---|
| 0.1 | 1.8 | 0.5 | 1.2 | 0.5 |
| 0.09 | 1.8085 | 0.50305 | 1.2022 | 0.50305 |
| 0.08 | 1.8171 | 0.50611 | 1.2044 | 0.50611 |
| 0.07 | 1.8257 | 0.5092 | 1.2066 | 0.5092 |
| 0.06 | 1.8343 | 0.51231 | 1.2087 | 0.51231 |
| 0.05 | 1.8429 | 0.51544 | 1.2109 | 0.51544 |
| 0.04 | 1.8516 | 0.51859 | 1.2131 | 0.51859 |
| 0.03 | 1.8604 | 0.52176 | 1.2152 | 0.52176 |
| 0.02 | 1.8691 | 0.52495 | 1.2174 | 0.52495 |
| 0.01 | 1.878 | 0.52816 | 1.2195 | 0.52816 |
| 0.009 | 1.8788 | 0.52849 | 1.2198 | 0.52849 |
| 0.008 | 1.8797 | 0.52881 | 1.22 | 0.52881 |
| 0.007 | 1.8806 | 0.52913 | 1.2202 | 0.52913 |
| 0.006 | 1.8815 | 0.52946 | 1.2204 | 0.52946 |
| 0.005 | 1.8824 | 0.52978 | 1.2206 | 0.52978 |
| 0.004 | 1.8833 | 0.5301 | 1.2208 | 0.5301 |
| 0.003 | 1.8841 | 0.53043 | 1.2211 | 0.53043 |
| 0.002 | 1.885 | 0.53075 | 1.2213 | 0.53075 |
| 0.001 | 1.8859 | 0.53108 | 1.2215 | 0.53108 |

number of training examples should be larger than the ratio of the total number of free parameters in the neural network to the tolerable value of the mean square of the estimation error *[6]*. The neural network was trained using a random input sequences over the interval *[-1 1]* with time steps Δ as given in *table-I* and for a time interval of 60 second for each Δ value, considering Δ and gama-inverse values of both $y_p(t)$ and $u(t)$ as training input and $y_p(t), a, b, c$ & $d$ as target.

Thus the equation using trained neural network that will govern the behaviour of the plant is given by:

$y_p(t) = NN_{12,14}\,[\Delta, \gamma^{-1}y_p(t), \gamma^{-2}y_p(t), \gamma^{-1}u(t), \gamma^{-2}u(t)]$, $t=k\Delta$ for $\Delta\neq 0$; $k$ is a positive integer

The trained neural network model is simulated considering sinusoidal test input sequences $u(t)=\sin(2\pi t/200) + \sin(2\pi t/100)$, ), [ $t=k\Delta$; $\Delta\neq 0$; $k$ is a positive integer] for different values of Δ. Some the simulation results are shown in the *fig.4* to *fig.6* with sampling period 0.06, 0.01 and 0.005 second respectively for simulation intervals of 60 second each.

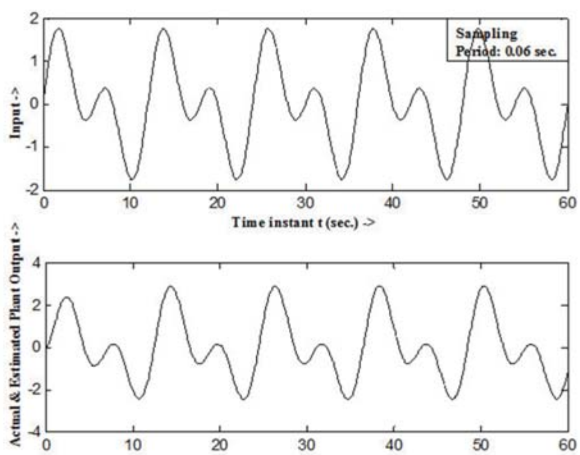


Fig4: Estimated Output of Neural Network Model for Δ=0.06Sec

The outputs of the plant and their estimates are shown by dark and light coloured lines respectively. It is seen that the output of the plant and identification model are almost indistinguishable i.e the identification model approximate the actual plant with very insignificant error.

Also the mean values and standard deviation values of estimated parameters generated by neural network in the simulation interval for different values of delta are given in *table-II* and *table-III* respectively. The values show that estimates are very close to actual parameter values given in *table-I* and converge towards continuous-time parameters as $\Delta \rightarrow 0$ .

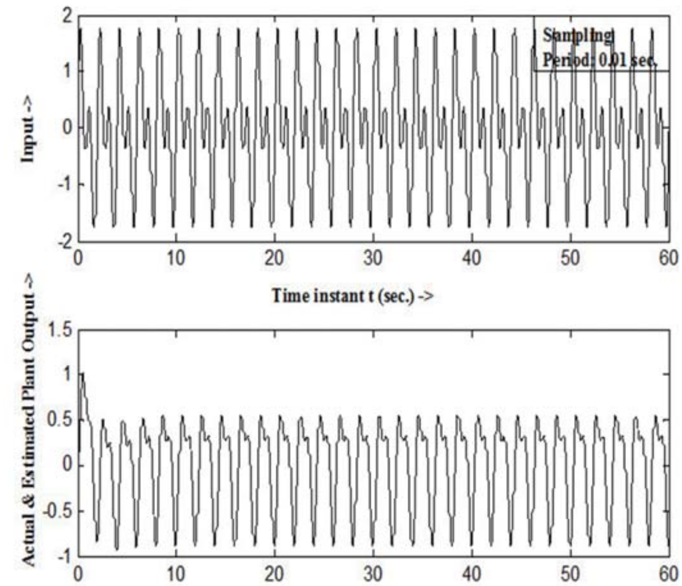


Fig5: Estimated Output of Neural Network Model for Δ=0.01Sec

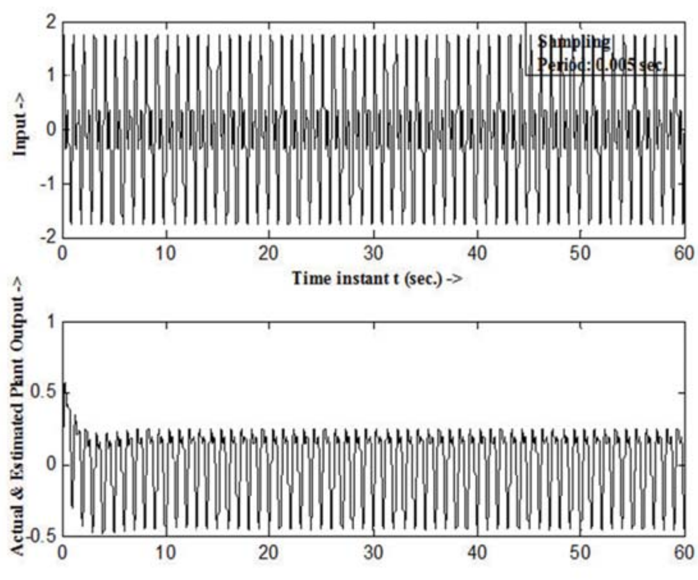


Fig6: Estimated Output of Neural Network Model for Δ=0.005Sec

TABLE-II: MEAN VALUES OF NEURAL NETWORK ESTIMATED COEFFICIENTS

| Δ | a | b | C | d |
|---|---|---|---|---|
| 0.1 | 1.8001 | 0.50003 | 1.2001 | 0.50003 |
| 0.08 | 1.817 | 0.50609 | 1.2044 | 0.50609 |
| 0.06 | 1.8343 | 0.51233 | 1.2088 | 0.51233 |
| 0.03 | 1.8604 | 0.52177 | 1.2152 | 0.52177 |
| 0.01 | 1.878 | 0.52817 | 1.2196 | 0.52817 |
| 0.007 | 1.8806 | 0.52911 | 1.2202 | 0.52911 |
| 0.005 | 1.8823 | 0.52973 | 1.2206 | 0.52973 |

TABLE-III: STANDARD DEVIATION VALUES OF NEURAL NETWORK ESTIMATED COEFFICIENTS

| Δ | a ($10^{-5}$) | b ($10^{-5}$) | c ($10^{-5}$) | d ($10^{-5}$) |
|---|---|---|---|---|
| 0.1 | 3.516 | 2.6858 | 3.0542 | 2.6858 |
| 0.08 | 3.2584 | 2.3426 | 2.6255 | 2.3426 |
| 0.06 | 2.822 | 1.5323 | 2.1172 | 1.5323 |
| 0.03 | 2.6051 | 1.2354 | 1.85 | 1.2354 |
| 0.01 | 2.2905 | 1.2944 | 1.0819 | 1.2944 |
| 0.007 | 3.1751 | 1.3846 | 1.4122 | 1.3846 |
| 0.005 | 3.5032 | 2.0632 | 1.9737 | 2.0632 |

## VI. CONCLUSION

Delta operator representation of dynamical system provides a robust framework in identification. The computational advantage of neural network is capitalized to develop structural framework using delta operator in which backpropagation algorithm is used in model identification. The mathematical formulation of this model structure encompasses linear time invariant systems in discrete time parameterization. The capabilities of the delta operator ensure robust numerical algorithm in computation and better numerical conditioning at high sampling frequency which is evident from the simulation result presented in the example for identification.

Therefore, from this work it is proved that neural network can be used as a candidate in modeling and identification of dynamical system in delta operator parameterization.

Also the proposed scheme can be applied for identification of nonlinear dynamical system; provided real time input-output data of the system for different sampling period is available to train the neural network and maximum order of the system is priori known.

## REFERENCES


[1] Richard H. Middleton And Graham C. Goodwin, 'Improved Finite Word Length Characteristics in Digital Control using Delta Operators', IEEE Transactions on Automatic Control. Vol. 31, No.11, Nov 1986.

[2] Middleton R. H. and Goodwin G. C., 'Digital Control and Estimation: A Unified Approach', Prentice Hall, New Jersey, 1990.

[3] Prasanta Sarkar, 'Reduced Order Modeling and Controller Design in Delta Domain', Phd Thesis, Department of Electrical Engg., Indian Institute of Technology, Kharagpur, Aug 2001.

[4] R. Hecht-Nielsen, 'Theory of the Back-Propagation Neural Network', Proceedings International Joint Conference on Neural Networks, pp, I-593-605, Jun 1989.

[5] K. Hornik, M. Stinchcombe and H. White, 'Multilayer Feedforward Networks are Universal Approximators', Neural Networks, Vol.2, pp 359-366, 1989.

[6] Simon Haykin, 'Neural Networks, A Comprehensive Foundation', Second Edition, Pearson Education, Inc.,1998.

[7] L. Ljung., 'System Identification: Theory for the user', 2nd Edition, Prentice Hall 1999.

[8] K. S. Narendra and K. Parthasarathy, 'Identification and Control of Dynamical Systems using Neural Networks', IEEE Transactions on Neural Networks, Vol. 1, pp. 4-27, Mar 1990.

[9] Chang Hu and Li Cao, 'A System Identification Method Based on Multilayer Perceptron and Model Extraction', LNCS 3174, pp. 218-223, 2004, Springer.

[10] Shigenobu Yamawaki and Lakhmi Jain, 'Robust System Identification Using Neural Networks', LNAI 3213 pp. 786-791, 2004, Springer.

[11] Francisco Cubillos and Gonzalo Acuna, 'Simulation Studies of On-Line Identification of Complex Processes with Neural Networks', LNCS 3972, pp. 808-814, 2006, Springer.

[12] Grace S. Wang and Fu-Kuo Huang, 'GA-Based Neural Network to Identification of Nonlinear Structural Systems', Part-III, LNCS 4493, pp.57-65, 2007.

[13] Nini Wang, Xiaodong Liu and Jianchuan Yin, 'A Hybrid Model of Partial Least Squares and RBF Neural Networks for System Identification', Part-I, LNCS 5263 pp 204-211,2008,Springer.

[14] Arit Thammano, Phongthap Ruxpakawong, 'Nonlinear Dynamic System Identification using Recurrent Neural Network with Multi-segment Piecewise-linear Connection Weights', Memetic Comp.(2010) 2:273-282, Springer.